

**Amendments to the Specification:**

Please amend the paragraph beginning at page 9, line 10 as follows:

Referring to the Drawing, wherein like numbers denote like parts throughout the several views, Figure 1 shows a high-level block diagram of a computer system 100, consistent with an embodiment of the invention. Computer system ~~[[100]]~~ 100 may comprise central processing unit (CPU) 101, main memory 102, terminal interface 103, data storage 104, and a network, e.g., Internet, interface ~~[[105]]~~ 105. The various devices communicate with each other via internal communications bus 110. CPU 101 is a general-purpose programmable processor, executing instructions stored in memory 102; while a single CPU is shown in Figure 1, it should be understood that computer systems having multiple CPUs could be used. Memory 102 is a random-access semiconductor memory for storing data and programs; memory is shown conceptually as a single monolithic entity but it is well known that memory is often arranged in a hierarchy of caches and other memory devices. Operating system 120 and applications 122 reside in memory 102. Operating system 120 provides, *inter alia*, functions such as device interfaces, management of memory pages, management of multiple tasks, etc. as is known in the art. Examples of such operating systems may include UNIX, WINDOWS-based, OS/400, etc. Applications 122 may include legacy applications and if it includes a server software application, network interface 105 may interact with the server software application ~~[[122]]~~ to enable computer system 100 to be a network server.

Please amend the paragraph beginning at page 13, line 1 as follows:

Client system 300 is a device separate from computer 100 that can access legacy applications which reside and run on the computer 100. Client system 300 may be a personal computer system or a larger computer system such as a server, or a smaller computer system,

such as notebook or laptop computer. Finally, client system 300 need not be a computer at all, but preferably is a simpler appliance-like client device with less memory such as a network terminal, a thin client, a terminal-like ~~devices~~ device, a voice response unit, etc. The convergence of computing, telecommunications and consumer electronics is causing a tremendous growth in the number and variety of pervasive mobile devices as clients 300. This mobile architecture enables the multitude of clients 300 including laptops, sub-notebooks, handheld computers, such as personal digital assistants and companion devices, and mobile appliances, such as smartphones, pagers, simple messaging devices and wearable devices. Thus when the client system 300 is a mobile device, a display adapter and network interface has a network user agent and supports a variety of multi-modal interfaces including traditional keyboard and mouse interfaces, small text screens, pen, touch screens, speech recognition, text-to-speech and other emerging technologies like wearable devices. A network user agent enables the use of the ~~computer~~s computer applications on its respective client 300. It is preferably intended that client system 300 include any electronic device which may interact with a network server 200 through the network user agent, such as a web browser, to access a legacy or other applications residing on the computer system 100. Such special-purpose devices for accessing the world wide web, such as an Internet access box for a television set, or a portable wireless web accessing device, which can implement a user agent for the purpose of invoking and executing an application are also intended to be within the scope of a client system 300. The network user agent could be implemented by control circuitry through the use of logic gate, programmable logic devices, or other hardware components in lieu of a processor-based system.

Please amend the paragraph beginning at page 21, line 12 as follows:

Typically, the display file source contains more information than the legacy application

data stream about the user interface. By converting the display file source at development rather than converting the application data stream during runtime, a better, more usable web page results. For example, the legacy application data stream may not include any information about which function keys are enabled for a particular record format. At runtime conversion, the resulting web page would have to have buttons for all twenty-four, more or less, function keys. With this approach of converting the display file source at development, a record format that only defines two function keys can be supported.